AFCARS

Record Identifier Encryption/Decryption Utility: C Version

Bulletin #10

May 26, 1995

---

1 INTRODUCTION

1.1 Overview

This technical bulletin provides guidance for installing and executing the AFCARS Record Identifier Encryption/Decryption Utility: C Version. The utility follows all guidelines listed for encryption software development found in **AFCARS Technical Bulletin #9: Encryption Guidelines and Algorithms**. It is designed to insure the confidentiality of State AFCARS data submissions; ACF will be **unable** to decode the encrypted record identifier. This utility consists of three programs which perform different functions:

1) One program will encrypt the client record identifiers in a State's properly formatted and concatenated AFCARS submission file **with delimiters**[1];

2) A second program will decrypt the record identifiers that have been encrypted using the AFCARS Record Encryption/Decryption Utility; and

3) A third program will decrypt the record identifiers in any report that is sent to the State that includes record identifiers that have been encrypted with the AFCARS Record Identifier Encryption/Decryption Utility, such as the detailed error reports that are sent back to the States after each submission.

IMPORTANT NOTICE: The AFCARS Record Identifier Encryption/Decryption Utility will process only ASCII formatted files as specified in the AFCARS Final Rule and **will not work for files created in EBCDIC and/or other non-ASCII formats.**

1.2 Utility Files

The utility is provided on a 3½", double-sided, high density diskette. To increase options to the States, the utility is available in two common languages: C and COBOL. This Bulletin gives instructions for using the C version. The enclosed diskette includes the source code and PC-compatible executables in the C language. Following is a complete listing of the diskette files:

---

[1] For instructions on how to properly format an AFCARS submission file with delimiters, see "Technical Bulletin #2: AFCARS File Format."

File Purpose	File Name
Encrypt Record Identifier (Executable)	ENCR_C.EXE
Decrypt Record Identifier (Executable)	DECR_C.EXE
Decrypt Standard Format Reports (Executable)	RDCR_C.EXE
Encrypt Record Identifier (Source Code)	ENCR_C.C
Decrypt Record Identifier (Source Code)	DECR_C.C
Decrypt Standard Format Reports (Source Code)	RDCR_C.C

1.3 Options for Using the Utility

ACF is providing both source code and executables to allow the States to select their own approach for encrypting record identifiers. Three options are listed below:

1) *Use the executables.* This is the easiest option. Simply copy the three executables to the PC where the AFCARS data submission is stored and execute them. The executables will only run on a PC under DOS. An authorized individual must enter the encryption/decryption key each time the program is executed.

2) *Use the source code.* This option provides the flexibility to execute the utility on different computer platforms. The source code can be copied to a PC or mainframe. The code must then be compiled and executed with the key entered at run-time. Additionally, the code may be edited by hard-coding the key directly into the source program prior to compilation. If the key is hard-coded, the utility will not prompt for the key when the program is executed.

3) *Use both executables and source code.* Options 1 and 2 may be combined. For example, the encryption/decryption key may be hard-coded into the source code of the encryption program, but the executables might be used for the file decryption and report decryption programs. An advantage of this option is that an operator would not be required to remember the key each time the encryption program is executed. However, an authorized individual would have to have access to the key to decrypt the record identifiers, thus adding an extra measure of security. If this option is selected, use the executables that were written in the same language as the modified source code.

1.4 Selecting either C or COBOL

If you decide to use source code, you have two options. You may either choose the C code enclosed or request COBOL code which is available from the AFCARS Technical Assistance Liaisons. This decision is determined by the State programming resources and available compilers. However, if programming resources are available in either language, the following factors should be considered:

1) *The COBOL version can be modified more extensively.* While the source code in both languages provides the opportunity to hard-code a key into the programs, the COBOL code permits the setting of an additional parameter for the algorithm.

2) *The C programs require fewer parameters at run-time.* The COBOL program requires two parameters entered at run-time while the C version requires only one.

1.5 The Encryption/Decryption Key

Regardless of which option is selected, authorized State personnel must agree precisely what the key will be in accordance with the guidelines and specifications outlined in **AFCARS Technical Bulletin #9: Encryption Guidelines and Algorithms** to determine how the record identifiers are encrypted. The key can be any 12 character combination of letters and/or numbers. However, it is imperative that the authorized personnel responsible for the key remember that:

1) The **exact** same key must be used for encrypting and decrypting files.

2) The key must be **exactly** 12 characters long.

3) The key is case sensitive. For example, "THISISTHEKEY" is not the same as "thisisthekey" or "Thisisthekey".

Since the key can be used to decrypt your AFCARS data submission, it must be **safeguarded against unauthorized access**. The key must also be **protected against loss**. If the key is lost, ACF will not be able to assist in decrypting the record identifiers since the key is determined by the State and only known to authorized State personnel.

If Option 1, "Use the executables," is selected for these programs, please follow the specific directions in Section 2 of this document. If using the source code, please follow the specific directions in Section 3 of this document.

2 USING ONLY THE EXECUTABLES - OPTION 1

2.1 Installing the Software

The enclosed 3½" diskette contains ready-to-use executable files for the file encryption, file decryption, and report decryption programs. Copy the files to an IBM 386 (or higher) compatible personal computer running DOS 3.0 or higher.

The executable files are as follows:

File Purpose	C File Name
Encrypting Record Identifier	ENCR_C.EXE
Decrypting Record Identifier	DECR_C.EXE
Decrypting Standard Format Reports	RDCR_C.EXE

2.2 Encrypting State Client Files

IMPORTANT NOTICE: The AFCARS Record Identifier Encryption/Decryption Utility will process only ASCII formatted files as specified in the AFCARS Final Rule and **will not work for files created in EBCDIC and/or other non-ASCII formats.**

Execute the encrypt command with one parameter - the name of the AFCARS detail file containing the State client identifiers. For example, if the AFCARS detail file name is "FC_DET.DOC", the command would be:

ENCR_C(space)FC_DET.DOC

The program will prompt for the encryption key. Type the encryption key and press <*Enter*>.

The file "OUTFILE" will be generated. The data in "OUTFILE" is unaltered; only the record identifiers will be encrypted.

2.3 Decrypting the State Client Files

A decryption routine is included to allow States to decode the record identifiers. This utility is included as a safeguard. If the AFCARS submission file is inadvertently lost, this utility can recreate it using the file generated by the encryption utility.

Execute the decrypt command with the name of the AFCARS detail file with the encrypted record identifiers. For example, if the input encrypted file name is "OUTFILE," the command would be:

DECR_C(space)OUTFILE

The program will prompt for the key. MAKE SURE TO USE EXACTLY THE SAME KEY that was used when the identifiers in the original file were encrypted using the AFCARS Record Identifier Encryption/Decryption Utility. Type the key and press <*Enter*>. A file named "DECRYPT.OUT" will be generated; it will be identical to the file that was originally encrypted.

2.4 Decrypting the AFCARS Detail Reports

Another important use for the AFCARS Record Identifier Encryption/Decryption Utility is to decrypt

the Detailed AFCARS Reports that are returned to the State by the Federal AFCARS system. These reports enumerate all errors that are subject to penalty in your submission file. States receive two of these report files for each AFCARS submission: one for foster care and one for adoption.

Since these files will be returned with the encrypted record identifiers, States will not be able to match them to the records in the State data system until the decryption routine is executed.

Execute the report decryption command with the name of the AFCARS Detail Report. For example, if the name of the report is "FC_RPT.DOC", the command would be:

RDCR_C(space)FC_RPT.DOC

The program will prompt for the key. MAKE SURE TO USE EXACTLY THE SAME KEY that was used when the identifiers in the original file were encrypted using the AFCARS Record number encryption utility Identifier Encryption/Decryption Utility. Type the key and press *<Enter>*. A file named "OUTPUT.DOC" will be generated; it will be identical to "FC_RPT.DOC" except that each record will have the State client identifier in place of the encrypted record identifier.

3 MODIFYING THE SOURCE CODE - OPTION 2

3.1 Installing the Software

To modify the source code for any of the three programs copy those files to a personal computer or mainframe. The file names for the source code files written in C are:

File Purpose	C Source Code File Name
Encrypting Record Identifier	ENCR_C.C
Decrypting Record Identifier	DECR_C.C
Decrypting Standard Format Reports	RDCR_C.C

After installing the source code, use a text editor to make changes to the source code. The following section explains what changes to make.

3.2 Modifying the C Source Code

The code may be modified to include the key. If you choose not to make this modification, you will be required to enter an encryption key manually at run-time.

Ordinarily, when executing the programs, the program will prompt for the key. Hard-coding an encryption key eliminates this prompt. The key *must* be 12 characters long. For example, **thisisthekey** is acceptable. (This example is for illustration purposes only; do not use this key when executing this utility.)

Figure 1 contains the C code that appears in all three source code files that needs to be modified. Modifications are **shaded**.

Original Code	Modified Code
/* ** ** This is what requires MODIFICATION ** Please note that source code will be shipped without ** any key at all. The following comment will be shipped ** with the source code. ** ** NOTE: You must enter the KEY into the variable "inputline" ** (a 12-character string) in the strcpy function call ** if you want the program to execute properly! */ strcpy(inputline,""); /* ** And you will comment out the following code, from here... */ printf("Enter the key: "); if (gets(inputline) == NULL) { fprintf(stderr, "You must supply a key for this program to run!\n"); fprintf(stderr, "Terminating...\n"); exit(-1); } if (strlen(inputline) != 12) { fprintf(stderr, "You must supply a key with exactly 12 characters.\n"); fprintf(stderr, "Your key has %d characters.\n", strlen(inputline)); fprintf(stderr, "Terminating...\n"); exit(-1); } /* ** To here... */	/* ** ** This is what requires MODIFICATION ** Please note that source code will be shipped without ** any key at all. The following comment will be shipped ** with the source code. ** ** NOTE: You must enter the KEY into the variable "inputline" ** (a 12-character string) in the strcpy function call ** if you want the program to execute properly! */ strcpy(inputline,"**thisisthekey**"); /* ** And you will comment out the following code, from here... */ **/*** printf("Enter the key: "); if (gets(inputline) == NULL) { fprintf(stderr, "You must supply a key for this program to run!\n"); fprintf(stderr, "Terminating...\n"); exit(-1); } if (strlen(inputline) != 12) { fprintf(stderr, "You must supply a key with exactly 12 characters.\n"); fprintf(stderr, "Your key has %d characters.\n", strlen(inputline)); fprintf(stderr, "Terminating...\n"); exit(-1); } ***/*** /* ** To here... */

FIGURE 1

Please note that while this code appears in all three C source code files, it is not necessary to modify it in all three files. For example, you may choose to hard-code the key only into the encrypt program, while not modifying the source code in the file decrypt or report decrypt programs. The advantage of this approach is that an operator would not need the key to execute the encryption program, but an authorized State individual will have to have access to the key to decrypt the record identifiers, thus adding an extra measure of security.

3.3 Compiling the C Source Code

After modifications are made, compile the source code using a C compiler to create an executable(s). **Compile all modified files**.

While the name of the executable(s) will not affect performance, we recommend the following naming conventions.

File Purpose	Executable Name
Encrypting Record Identifier	ENCR_C.EXE
Decrypting Record Identifier	DECR_C.EXE
Decrypting Standard Format Reports	RDCR_C.EXE

3.4 Encrypting the State Client Files

IMPORTANT NOTICE: The AFCARS Record Identifier Encryption/Decryption Utility will process only ASCII formatted files as specified in the AFCARS Final Rule and **will not work for files created in EBCDIC and/or other non-ASCII formats.** Operational instructions for C follow:

Execute the encrypt command with one parameter - the name of the AFCARS detail file containing the State client identifiers. For example, if the AFCARS detail file name is "FC_DET.DOC", the command would be:

ENCR_C FC_DET.DOC

If the encryption key is not hard-coded into the source code, the operator will be prompted for the key. Type the key and press <*Enter*>. If the code is modified to include the encryption key, no action is required. A file named "OUTFILE" will be generated. The data in "OUTFILE" is unaltered; only the record identifiers will be encrypted.

3.5 Decrypting the State Client Files

A decrypting routine is included to allow States to decode the record identifiers. This utility is included as a safeguard. If the AFCARS submission file is inadvertently lost, this utility can recreate it using the file generated by the encryption utility.

Execute the decrypt command with the name of the AFCARS detail file with the encrypted record numbers. For example, if the input encrypted file name is "OUTFILE" the command would be:

DECR_C(space)OUTFILE

The program will prompt for the key. MAKE SURE TO USE EXACTLY THE SAME KEY that was used when the identifiers in the original file were encrypted using the AFCARS Record Identifier Encryption/Decryption Utility. Type the key and press <*Enter*>. A file named "DECRYPT.OUT" will be generated; it will be identical to the file that was originally encrypted.

3.6 Decrypting the AFCARS Detail Reports

Execute the report decryption command with the name of the AFCARS Detail Report. For example, if the name of the report is "FC_RPT.DOC," the command would be:

RDCR_C(space)FC_RPT.DOC

The program will prompt for the key. MAKE SURE TO USE EXACTLY THE SAME KEY that was used when the identifiers in the original file were encrypted using the AFCARS Record Identifier Encryption/Decryption Utility. Type the key and press <*Enter*>. A file named "OUTPUT.DOC" will be generated; it will be identical to FC_RPT.DOC except that each record will have the State client identifier in place of the encrypted record identifier.

4 ADDITIONAL INFORMATION

If additional information is required on installing and executing the AFCARS Record Identifier Encryption/Decryption Utility, please contact your AFCARS Technical Assistance Liaison.